IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Applicant(s):	Masek et al.	**Conf. No.**:	5987
Serial No.:	10/670,898	**Art. Unit**:	2193
Filed:	09/25/2003	**Examiner**:	Mitchell, Jason D.
Title:	METHOD, SYSTEM, AND PROGRAM PRODUCT FOR TESTING A SERVER USING A REENTRANT TEST APPLICATION	**Docket. No.**:	LOT920030024US1 (IBML-0027)

Mail Stop Reply Brief - Patents
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

REPLY BRIEF

Sir:

Introductory Comments:

This paper is being filed in response to the Examiner's Answer dated 28 July 2011. Appellants responds as follows:

Remarks:

Appellants respectfully submit that the Examiner has misstated facts relevant to the rejections, and in so doing, confuses the issues to be considered on Appeal. Accordingly, Appellants respond.

Appellants note that the rejection of claims 1-9, 11-18, and 20-26 under 35 U.S.C. 112, first paragraph, have been withdrawn.

Regarding the Examiner's response to the rejection of claims 1-5, 7-9, 11-14, 16-18, 20-23, and 25-26 under 35 USC 103(a) (Examiner's Answer, Page 12-13), Appellants reassert the arguments from the Appeal Brief. In addition, the Examiner states that the limitation of "instantiating, via the test script, a plurality of instances of the test application using threads, wherein the instantiating and execution of each of the plurality of instances of the test application occur within a single process," is taught by Duggan and Dinker. The Examiner admits that Duggan does not explicitly disclose that the plurality of instances of the test application using threads occurs within a single process. (Examiner's Answer, Page 13). Rather, the Examiner relies on Dinker. As previously argued, every disclosed embodiment of Dinker specifies several multi-threaded test processes by implementing a test cluster 100. (Dinker, Para. 31). The Examiner's assertion that Dinker only prefers a plurality is incorrect, as Dinker never discloses a single process. Appellants interpret Dinker's statement that "some embodiments may encounter less thread starvation that if the same number of clients were simulated by a single multi-threaded process" as admitting that the thread starvation may be less, or it may not, but no embodiment teaches actually utilizing a single process. As such, Dinker is being speculative, but does in fact teach away from a single process. For instance, Dinker specifies that it may result in less thread starvation, but also that each of the test agents may be executed on a lower performance computer for a lower cost than trying to execute them in a single multi-threaded process. (Dinker, Para. 31). As such, Dinker does teach away from using a single process.

Further, the Examiner posits that as Duggan does not teach generating multiple processes, it is possible that Duggan teaches a single process. (Examiner's Answer, Page 15). However, Appellants assert that as the Examiner states, it is improper to read a disclosure which does not describe a limitation as requiring such a limitation. Since Duggan also fails to describe the plurality of instances of the test application running in a single process, it is improper to read Duggan as running them in a single process. Accordingly, as clarified based on the Examiner's remarks, Appellants assert that all of the features of the claim are not disclosed within Duggan in view of Dinker for claim 1, and similarly the other independent claims.

Regarding the Examiner's response to the limitation of "sharing all services and memory space which are exclusively dedicated to the single process with others of the plurality of instances," Appellants submit that this limitation is not very broad. (Examiner's Answer, Page 16). To one skilled in the art, sharing the services and memory space which are exclusively dedicated to the single process means that the services and memory are used for only this testing. Thus, the distinction between "share information using memory" and "share a memory space" (Examiner's Answer, Page 16) is inherently clear. To share information using memory, and memory of a computing device may be used. Sharing memory space as claimed means that the memory is used only for the testing, and thus there is a dedicated memory space to be shared amongst the process. Partamian's sharing of information using memory does not define what memory is utilized, and could vary by using any and all memory in a system. Further, Duggan does not teach sharing memory simply due to disclosing that a test operator can stress an application program. (Examiner's Answer, Page 17, citing Duggan, Col. 21, Lines 59-62). Rather, Appellants argued that the instant application discloses testing based specifically on the load generated, but also that the memory space is shared. For instance, "the present invention will instantiate multiple instances of the test application to test the server application. As such, multiple test applications will be running in the same process, sharing the same services and memory space." (Instant Application, Pages 5-6). No such disclosure exists in Duggan, which was the point made in the Appeal Brief. Appellants disclosed testing the load generated and sharing the memory space to effectively do so. Accordingly, Appellants maintain that the cited combination fails to teach or suggest each and every feature of the independent claims.

Regarding the Examiner's response to the limitation of "... identifying application protocol interfaces (APIs) ..., prior to the instantiating step...[and] providing a test script capable of invoking the APIs ..." Appellants, assert that regardless of Visual Basic being a program or a programming language, Appellants assert that the disclosure of Firth simply teaches that "Internet API functions can be used from programming languages (e.g., Visual Basic® by Microsoft)." (Examiner's Answer, Page 18, citing Firth, Col. 20, lines 50-52). However, this reference disclosing that Visual Basic can be used for an API still fails to make any reference to test applications or any related information. Further, Duggan describing a programming language with no disclosure of the use of APIs is not remedied by Firth's disclosure that Visual Basic can be used to call APIs. Appellants respectfully submit that neither reference teaches the use of APIs within a testing environment. Accordingly, Appellants maintain that the cited combination fails to teach or suggest each and every feature of the independent claims.

Regarding the rest of the Examiner's Answer, it relies on the features discussed above. Accordingly, Appellants rely on the arguments above as well as those presented in the Appeal Brief.

Appellants respectfully submit that the cited combination does not teach or suggest all of the claim limitations of the independent claims. Regarding the dependent claims, Appellants hereby incorporate the above arguments with respect to the independent claims. Accordingly, Appellants maintain that the dependent claims are patentable based on the arguments above as well as their own features.

Accordingly, Appellants submit that all pending claims are allowable because Duggan, Dinker, Partamian, and Firth fail to teach or suggest each and every feature of the claims as required by 35 U.S.C. 103(a).

Respectfully submitted,

/Nathan B. Davis/

Nathan B. Davis
Reg. No. 67,474

Date: 28 September 2011

Hoffman Warnick LLC
75 State Street, 14th Floor
Albany, New York 12207
Phone: (518) 449-0044
Fax: (518) 449-0047